

**LAPORAN PRAKTIKUM  
ALGORITMA PEMROGRAMAN**

**“PERULANGAN FOR DAN NESTED FOR”**

**DISUSUN OLEH:  
ARYAHIYAHUL FIKRA  
2511532026**

**DOSEN PENGAMPU:  
Dr. WAHYUDI, S.T, M.T  
ASISTEN PRAKTIKUM:  
RAHMAD DWIRIZKI**



**DEPARTEMEN INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
PROGRAM STUDI INFORMATIKA  
UNIVERSITAS ANDALAS  
2025**

## **KATA PENGANTAR**

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa karena atas rahmat dan karunia-Nya, Laporan praktikum ini dapat diselesaikan dengan baik. Laporan praktikum ini disusun sebagai salah satu tugas dalam rangka memenuhi tugas yang diberikan oleh dosen dan juga asisten praktikum.

Padang, 03 Oktober 2025

Aryahiyahul fikra

## **DAFTAR ISI**

KATA PENGANTAR .....	ii
DAFTAR ISI.....	iii
BAB I.....	1
PENDAHULUAN .....	1
1.1    Latar Belakang .....	1
1.2    Tujuan .....	2
1.3    Manfaat Pratikum.....	2
BAB II.....	3
PEMBAHASAN .....	3
2.1    Pengertian PerulanganFor .....	3
2.2    Pengertian NestedFor.....	3
2.3    Pemograman Pekan 5.....	4
2.3.1    Perulangan For 1 .....	4
2.3.2    Perulangan For 2.....	4
2.3.3    Perulangan For 3.....	5
2.3.4    Perulangan For 4.....	6
2.3.5    Nested For 0.....	6
2.3.6    Nested For 1 .....	7
2.3.7    Nested for 2.....	8
BAB III .....	9
PENUTUP .....	9
3.1    Kesimpulan .....	9
3.2    Saran .....	9
DAFTAR PUSTAKA .....	10

## BAB I

### PENDAHULUAN

#### 1.1 Latar Belakang

Perkembangan teknologi informasi yang pesat dalam beberapa dekade terakhir telah memberikan dampak besar terhadap berbagai aspek kehidupan manusia, termasuk di bidang pendidikan, industri, dan komunikasi. Salah satu pilar utama dalam perkembangan teknologi tersebut adalah **pemrograman komputer**. Di antara banyak bahasa pemrograman yang ada, **Java** merupakan salah satu bahasa yang sangat populer dan banyak digunakan di berbagai belahan dunia.

Java adalah bahasa pemrograman tingkat tinggi yang bersifat **object-oriented**, dikembangkan oleh Sun Microsystems pada tahun 1995, dan kini dikelola oleh Oracle. Keunggulan utama Java terletak pada kemampuannya untuk berjalan di berbagai platform (platform-independent) berkat konsep **Write Once, Run Anywhere (WORA)** yang dimilikinya. Hal ini memungkinkan program yang dibuat dalam Java untuk dijalankan di berbagai sistem operasi tanpa perlu diubah ulang.

Selain itu, Java memiliki sintaks yang relatif mudah dipahami bagi pemula, dokumentasi yang lengkap, dan komunitas pengguna yang besar, sehingga sangat mendukung proses belajar dan pengembangan perangkat lunak. Java juga menjadi bahasa utama dalam pengembangan berbagai aplikasi, seperti aplikasi desktop, mobile (Android), web, hingga sistem enterprise berskala besar.

Dengan melihat pentingnya Java dalam dunia pemrograman dan aplikasinya yang luas, sangatlah relevan untuk mempelajari dasar-dasar bahasa pemrograman Java, baik dari segi teori maupun praktik. Oleh karena itu, laporan ini disusun untuk memberikan pemahaman yang lebih mendalam tentang bahasa Java, struktur dasarnya, serta penerapan konsep pemrograman berorientasi objek dalam bahasa tersebut.

### **1.2 Tujuan**

1. Membantu mahasiswa memahami konsep dasar pemrograman Java melalui penerapan langsung.
2. Melatih kemampuan menulis, mengompilasi, dan mengeksekusi program dengan mengikuti aturan sintaksis Java.
3. Meningkatkan keterampilan dalam memecahkan masalah (problem solving) dengan pendekatan algoritmik
4. Membiasakan mahasiswa bekerja sistematis dalam menyusun laporan yang memuat analisis hasil praktikum
5. Menanamkan sikap teliti, disiplin, serta tanggung jawab dalam melaksanakan kegiatan laboratorium.

### **1.3 Manfaat Praktikum**

1. Mahasiswa dapat memahami konsep dasar pemrograman secara praktis, bukan hanya teori.
2. Melatih keterampilan logika dan analisis dalam menyusun serta mengeksekusi kode.
3. Memberikan pengalaman langsung dalam menggunakan bahasa Java sebagai salah satu bahasa pemrograman populer.
4. Menjadi bahan awal untuk memahami materi pemrograman yang lebih kompleks pada praktikum berikutnya.
5. Membiasakan mahasiswa dengan proses debugging dan pemecahan masalah dalam pemrograman.

## **BAB II**

### **PEMBAHASAN**

#### **2.1 Pengertian PerulanganFor**

Perulangan for adalah jenis perulangan yang digunakan untuk mengulang eksekusi blok kode dalam jumlah yang sudah diketahui.

1. Perulangan for terdiri dari tiga bagian utama:
2. Inisialisasi – menentukan nilai awal variabel.
3. Kondisi – menentukan kapan perulangan berhenti.
4. Iterasi – menentukan bagaimana nilai variabel berubah setiap kali perulangan dijalankan.

#### **2.2 Pengertian NestedFor**

Nested for adalah perulangan for yang diletakkan di dalam perulangan for lainnya. Dengan kata lain, satu perulangan berada di dalam perulangan lain.

1. Struktur ini banyak digunakan dalam pemrograman untuk:
2. Mengelola data dua dimensi (misalnya matriks atau tabel),
3. Membuat pola teks atau grafik sederhana,
4. Melakukan perhitungan bersarang (nested computation)

## 2.3 Pemograman Pekan 5

### 2.3.1 Perulangan For 1

```
1 package pekan5;
2
3 public class PerulanganFor1 {
4
5     public static void main(String[] args) {
6         for (int i = 1; i <= 10; i++){
7             System.out.println( i );
8         }
9     }
10 }
11
12 }
13 }
```

Kode Pemograman 2.3.1

Pada pemograman ini, program perulangan for yang membariskan angka dari 1 – 10 sesuai dengan int i tersebut.

Output dari pemograman diatas

```
1
2
3
4
5
6
7
8
9
10
```

### 2.3.2 Perulangan For 2

```
1 package pekan5;
2
3 public class Perulangan2 {
4
5     public static void main(String[] args) {
6         for( int i= 1; i <= 10; i++) {
7             System.out.print(i+" ");
8         }
9     }
10 }
11
12 }
13 }
```

### Kode Pemograman 2.3.2

Pada pemograman ini hampir mirip sama kode pemograman 2.3.1, yang berbeda adalah dari outputnya yang pada pemograman 2.3.1 mengarah kebawah pada pemograman ini mengarah ke kanan seperti pada foto output dibawah.

Output dari pemograman diatas

```
1 2 3 4 5 6 7 8 9 10
```

### 2.3.3 Perulangan For 3

```
1 package pekan5;
2
3 public class PerulanganFor3 {
4
5     public static void main(String[] args) {
6         int jumlah = 0;
7         for(int i=1; i<=10;i++) {
8             System.out.print(i);
9             jumlah+= jumlah+i;
10            if (i<10) {
11                System.out.print(" + ");
12            }
13        }
14        System.out.println();
15        System.out.println("jumlah = " +jumlah);
16    }
17}
18}
19}
20}
21}
```

### Kode Pemograman 2.3.3

Pada pemograman diatas perulanngan for yang berbeda dari perulangan for sebelumnya. Perulangan for kali ini yaitu menggunakan if tambahan dan juga outputnya menjumlahkan angka 1 – 10.

Output dari pemograman tersebut

```
1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10
jumlah = 55
```

### 2.3.4 Perulangan For 4

```

1 package pekan5;
2
3 import java.util.Scanner;
4
5 public class PerulanganFor4 {
6
7     public static void main(String[] args) {
8         int jumlah =0;
9         int batas;
10        Scanner input = new Scanner(System.in);
11        System.out.print("Masukkan nilai batas = ");
12        batas= input.nextInt();
13        input.close();
14        for(int i=1; i <=batas;i++ ) {
15            System.out.print(i);
16            jumlah = jumlah+i;
17            if(i<batas) {
18                System.out.print(" + ");
19            }else {
20                System.out.print(" = ");
21            }
22        }
23        System.out.println(jumlah);
24
25    }
26}
27
28}
29

```

Kode Pemograman 2.3.4

Pada pemograman diatas kali ini pemograman tersebut menggunakan scanner yang berguna untuk memasukkan angka yang diinginkan dan juga menggunakan if else.

Output dari pemograman diatas

```

Masukkan nilai batas = 15
1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 + 11 + 12 + 13 + 14 + 15 = 120

```

### 2.3.5 Nested For 0

```

1 package pekan5;
2
3 public class NestedFor0 {
4
5     public static void main(String[] args) {
6         for (int line = 1; line <= 5; line++) {
7             for (int j = 1; j <= (-1 * line + 5); j++) {
8                 System.out.print(".");
9             }
10            System.out.print(line);
11            System.out.println();
12        }
13    }
14}
15
16
17
18
19
20

```

Kode Pemograman 2.3.5

Pada pemograman diatas berbeda dengan perulangan for, kali ini pemograman yang mengurutkan angka kebawah akan tetapi terdapat titik – titik yang membedakan dengan perulangan for pertama.

Ouput dari pemograman diatas

```
....1
...2
..3
.4
5
```

### 2.3.6 Nested For 1

```
1 package pekan5;
2
3 public class NestedFor1 {
4
5     public static void main(String[] args) {
6         for (int i = 1; i <= 5; i++) {
7             for (int j = 1; j <=5; j++) {
8                 System.out.print("*");
9             }
10            System.out.println();
11            // to end the line
12        }
13    }
14}
15
16
17 }
```

Kode Pemograman 2.3.6

Pada pemograman diatas menggunakan konsep nested for atau perulangan bersarang, terdapat dua perulangan for Perulangan luar mengatur jumlah baris yang akan ditampilkan, sedangkan perulangan dalam mengatur jumlah kolom bintang (\*) yang muncul di setiap baris.

Ouput dari pemograman diatas

```
*****
*****
*****
*****
*****
```

### 2.3.7 Nested for 2

```
1 package pekan5;
2
3 public class NestedFor2 {
4
5     public static void main(String[] args) {
6         for (int i = 0 ; i <= 5 ; i++) {
7             for (int j = 0 ; j <= 5; j++) {
8                 System.out.print(i+j+ " ");
9             }
10            System.out.println();
11            // to end the line
12        }
13    }
14 }
15 }
16 }
```

Kode Pemograman 2.3.7

Pada pemograman diatas terdapat dua perulangan for, Perulangan luar berfungsi untuk mengatur jumlah baris yang akan ditampilkan, sedangkan perulangan dalam berfungsi untuk mengatur jumlah kolom di setiap baris.

Ouput dari pemograman diatas,

```
0 1 2 3 4 5
1 2 3 4 5 6
2 3 4 5 6 7
3 4 5 6 7 8
4 5 6 7 8 9
5 6 7 8 9 10
```

## **BAB III**

### **PENUTUP**

#### **3.1 Kesimpulan**

Perulangan for merupakan struktur dasar pemrograman yang sangat penting untuk melakukan tugas berulang secara efisien. Dengan memahami cara kerja perulangan for dan nested for, programmer dapat menulis kode yang lebih ringkas, efisien, dan mudah dikelola.

Penggunaan nested for sangat berguna dalam situasi yang melibatkan data dua dimensi atau perhitungan bersarang. Namun, perlu diperhatikan kompleksitas waktu agar program tetap efisien..

#### **3.2 Saran**

Pemahaman konsep dasar perulangan sangat penting sebelum mempelajari algoritma yang lebih kompleks. Disarankan bagi pembaca untuk berlatih membuat berbagai contoh kasus menggunakan perulangan for dan nested for agar pemahaman lebih mendalam.

## DAFTAR PUSTAKA

Sumber Daring (website):

- [1] Abdul Kadir. (2013). Dasar Pemrograman Python. Yogyakarta: ANDI.
- [2] Tutorialspoint. (2024). “Python For Loop Tutorial.” [Online]. Available at: [https://www.tutorialspoint.com/python/for\\_loop.htm](https://www.tutorialspoint.com/python/for_loop.htm)
- [3] Wahana Komputer. (2019). Belajar Pemrograman dengan Python. Yogyakarta: Andi Publisher.